ANL/ALCF/ESP-17/11

Free Energy Landscapes of Membrane Transport Proteins

Technical Report for the ALCF Theta Early Science Program

Argonne Leadership Computing Facility

ALCF Early Science Program (ESP) Technical Report

ESP Technical Reports describe the code development, porting, and optimization done in preparing an ESP project's application code(s) for the next generation ALCF computer system. This report is for a project in the Theta ESP, preparing for the ALCF Theta computer system.

About Argonne National Laboratory
Argonne is a U.S. Department of Energy laboratory managed by UChicago Argonne, LLC
under contract DE-AC02-06CH11357. The Laboratory's main facility is outside Chicago, at
9700 South Cass Avenue, Argonne, Illinois 60439. For information about Argonne
and its pioneering science and technology programs, see www.anl.gov.

Disclaimer
This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor UChicago Argonne, LLC, nor any of their employees or officers, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of document authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof, Argonne National Laboratory, or UChicago Argonne, LLC.

ANL/ALCF/ESP-17/11

Free Energy Landscapes of Membrane Transport Proteins

Technical Report for the ALCF Theta Early Science Program

edited by
Timothy J. Williams and Ramesh Balakrishnan

Argonne Leadership Computing Facility

prepared by
Brian K. Radak, James C. Phillips, Wei Jiang, Sunhwan Jo, Laxmikant Kalé, Klaus Schulten, and Benoît Roux

September 2017

This work was supported in part by the Office of Science, U.S. Department of Energy, under Contract DE-AC02-06CH11357.

Free Energy Landscapes of Membrane Transport Proteins

Brian K. Radak[1], James C. Phillips[2], Wei Jiang[1], Sunhwan Jo[1], Laxmikant Kalé[4], Klaus Schulten[2], and Benoît Roux[*3]

[1] Leadership Computing Facility, Argonne National Laboratory, Argonne, IL
[2] Beckman Institute for Advanced Science and Technology, University of Illinois at Urbana–Champaign, Urbana, IL
[3] Department of Biochemistry & Molecular Biology, University of Chicago, Chicago, IL
[4] Department of Computer Science, University of Illinois at Urbana–Champaign, Urbana, IL

1 Introduction

Molecular dynamics (MD) is now a widespread tool for investigating biochemical and biomolecular systems, its prevalence, in no small part, being due to significant advances in computational hardware over the last few decades. Even non-experts can now routinely use MD for protein and nucleic acid structure refinement, studying conformational switching events, and investigating solvation effects due to ligand/drug binding. Nonetheless, the vast majority of simulations being done today utilize only rudimentary algorithmic approaches – so-called "brute force" MD – which only permit access to a small fraction of what the approach has to offer. This is largely because the core algorithm is simple, while advanced techniques can require everything from more sophisticated models and data structures to complicated on-the-fly analysis. A core goal of this early science project is to bring one such advanced MD approach into the broader arena of high-performance computing. In particular, the powerful method of constant-pH MD has long been a relegated to experienced users and specialized model systems. In this work we develop and implement a constant-pH MD algorithm in the NAMD simulation engine making it suitable for deployment on large-scale, next-generation supercomputers as well as ambitious, cutting-edge biological applications. We report, for the first time, constant-pH simulations of a membrane transport protein and use the results to analyze its free energy landscape for ion-selectivity.

2 Science Summary

Proteins are the molecular machinery of the cell, carrying out a broad range of functions from energy production and cargo transportation to cellular gating and signal transduction. These varied functionalities are possible because proteins are able to change between specific, usually well-defined, sets of shapes which often have chemically or mechanically distinct behaviors. Understanding these protein "structure-function" relationships is a core challenge in biochemistry. Computational

[*] roux@uchicago.edu

models are a powerful tool in these investigations because they provide an unambiguous picture of the biomolecular system under study and permit a detailed theoretical correspondence with experimental results as well as a pathway to generating highly specific chemical hypotheses for future testing. This closed loop of experimental-theoretical exploration, testing, and validation is now a cornerstone of numerous fields.

While molecular simulations can, in principle, capture and predict the intrinsic dynamics of molecules, this behavior is often too complicated, varied, or even random to be of general use. Instead, the machinery of statistical physics permits a simplification whereby these behaviors are reduced to a set of probabilistic outcomes governed not by *mechanical* forces, but by *thermodynamical* forces. Chief amongst these thermodynamic quantities is free energy. Computing free energies allows us to: 1) determine the relative population of various protein conformations (*i.e.*, shapes and structures), 2) study the ability of those conformations to perform a given task, and 3) infer the factors that affect the biochemical process in question.

2.1 Free Energy and pH

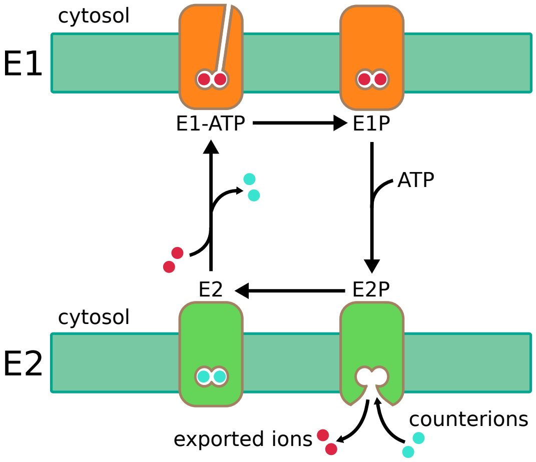

Figure 1: General schematic of the catalytic cycle for P-type ATPases. Transport and countertransport of ions (various species/stoichiometries are possible) is accompanied by interconversion of two main conformational states (E1 and E2). Adapted from Ref. [3].

One extremely important chemical factor is the pH of a solution. The pH quantifies the propensity for protons, which carry a positive charge, to bind or unbind from specific sites on a protein. The constant flux of protons causes local changes in the electrostatic forces on a protein and can elicit changes in conformation, binding affinity (*e.g.* to drug compounds), or even chemical reactivity. The free energy landscape of a protein is thus intrinsically linked to the pH of its environment which often has significant effects on both protein structure and function. Despite this clear importance, the vast majority of molecular simulations do not directly integrate pH as a fundamental parameter. This is partly because such an approach is not always necessary and requires additional, potentially difficult, considerations in the computational model. A primary goal of this work is to address this challenge.

2.2 P-type ATPases

Transmembrane domains of P-type ATPases are a broad class of transport proteins found in both prokaryotic and eukaryotic cells. In concert with soluble domains, these transmembrane proteins perform critical cell functions by consuming ATP in order to maintain distinct ionic gradients across various membranes [16, 3]. The function of P-type ATPases involves complex conformational transitions that couple ion transport across the membrane with ATP/ADP binding or unbinding. These transitions include cycling between at least four states which “alternatingly occlude” one side of the membrane while remaining open to one, the other, or neither [16] (Figure 1). Furthermore, the coupling with ion transport depends on *selectivity* for a particular type of ion (*e.g.* Na^+ versus

K^+), which is thought to be highly correlated with the protonation state of key parts of the protein. Probing models of these proteins under different pH conditions is a direct way to investigate these correlations and may provide insights into how different states establish ion selectivity and couple this into directional motion in the transport cycle.

3 Codes, Methods and Algorithms

This project utilizes and extends the NAMD simulation engine [21]. NAMD is primarily designed for all-atom and coarse-grained MD with a focus on biomolecular applications. The two main ingredients for this method are a particle-based integration scheme for the equations of motion (usually Newtonian or Langevin dynamics) and a molecular "force field" with critical reliance on a long-range pair potential for electrostatics. The performance bottlenecks for these are the timescale, Δt, on which the motion of particles can be numerically integrated with stability and the cost of computing the $\mathcal{O}(N^2)$ pair interactions in a system of N particles, respectively.

3.1 Molecular Dynamics in NAMD

NAMD, like nearly all other codes of it kind, addresses the numerical integration problem by using a Verlet-type multiple timestep integrator [2, 32] with holonomic constraints on high frequency degrees of freedom [25, 1]. The fastest degrees of freedom are thus made "rigid," while incrementally slower interactions are given longer timesteps. In all-atom models, the rigid degrees of freedom are almost exclusively covalent bonds with hydrogen atoms, especially those between oxygen and hydrogen in water molecules, which compose the bulk of biomolecular systems in solution. NAMD gives special consideration to water by using the SETTLE [19] algorithm and single instruction multiple data (SIMD) vectorization. The fast "floppy" degrees of freedom are then covalent and proximal non-covalent interactions, while the slowest degrees of freedom are those due to distal long-range interactions.

The state-of-the-art for biomolecular simulations of finite extent is to employ periodic "images" of a central cell in order to emulate physical interactions at larger length scales. Since nearly all contemporary atomistic models utilize point charge (or point multipole) electrostatics with long-range dependence, this requires some approximation to account for pair interactions between not only the particles in the primary cell, but also the image atoms in adjoining cells. Probably the most popular and widely used approximation is the particle-mesh Ewald [10, 11] (PME) summation technique, which exploits the imposed periodicity by using fast-Fourier transforms to compute interactions with a periodic "mesh" at large distances. Although NAMD also permits other electrostatic approximations [31, 12], the present work exclusively utilizes the $\mathcal{O}(N \log N)$ PME approach which is implemented in the Charm++ programming language (based on C++). Using Charm++ permits automated load-balancing and reliable scaling for systems of several orders of magnitude of size, as many as $\mathcal{O}(10^8)$.

3.2 Hybrid Nonequilibrium Molecular Dynamics/Monte Carlo

There has been considerable interest of late in hybrid techniques combining MD with conventional Metropolis Monte Carlo [18] (MC). Notably, a large number of methods from the literature, many of which were not widely used or properly appreciated, were recently gathered [20] under the aegis of the work theorems of Jarzynski [14] and Crooks [8, 9]. This theoretical apparatus was later refined and simplified for practical applications, an approach we refer to as hybrid nonequilibrium MD/MC (neMD/MC) [5, 7].

The essential principle behind neMD/MC is that statistical sampling from an *equilibrium* distribution (the primary quantities of interest in most biological applications) can be obtained from *nonequilibrium* trajectories in which the normally stochastic behavior is perturbed by a deterministic process controlled by the user. The algorithmic amendment that makes this possible is the addition of an accept/reject criterion based on the nonequilibrium work exerted during the trajectory. This approach is exceedingly general and can be applied to any coordinate of interest, including physical collective variables as well as thermodynamic parameters. The critical problem is in choosing the coordinate systematically so as to speed up statistics for the problem at hand. Unfortunately, a generic answer to this problem will not be presented here. Instead, we shall show how a specific parameterization of atomistic force fields can be used to sample a new, but specific, degree of freedom, the protonation state.

3.3 Constant-pH Molecular Dynamics in NAMD

In a broad sense, pH can be considered as a thermodynamic force which pushes and pulls protons from various sites on a molecule. Whenever a proton changes the site it occupies, the protonation state, and potentially the behavior of the molecule, also changes. This set of possible protonation states can be considered as a network, each with an overlapping set of structural characteristics, but also distinct differences. Conventional, force-field based MD is designed to investigate the behavior of a *single* protonation state – a task at which it has proven quite adept. The missing component for constant-pH MD is how to link the network of states together in a physically meaningful way (*i.e.,* by considering the pH "forces").

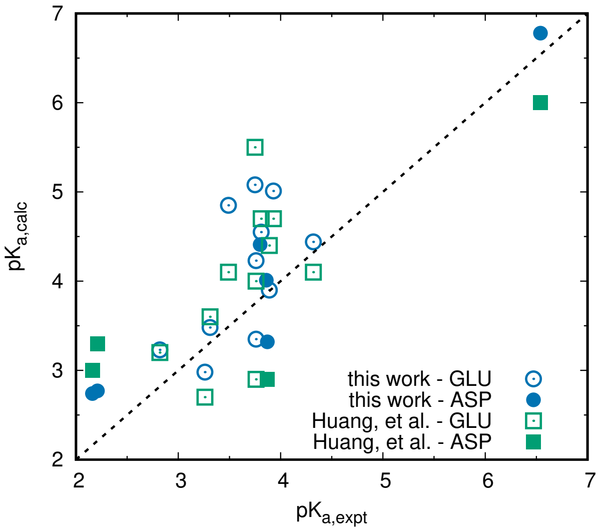

Figure 2: Benchmark simulations on a well-studied protein system show good agreement between the present implementation [23], experiment [4], and previous computational efforts [13].

The basic constant-pH scheme in NAMD is to alternately sample the protonation state and then the configuration space within that state. As originally proposed by Stern two decades ago [28], protonation state sampling is accomplished by neMD/MC, whereby a proton that is non-interacting (*i.e.,* its site is empty) has its interactions forcibly turned on (or vice-versa) in a time-dependent manner (*i.e.,* over the course of a short trajectory). Although this approach is exceedingly straightforward, certain nuances and practical aspects were not fully satisfactory until recently. In particular, the details of the nonequilibrium trajectories can frustrate or bias statistics [5] and correctly and accurately computing the work criterion requires diligent bookkeeping [7]. Furthermore, a naive implementation of the algorithm for systems with many protonation sites may be disastrously inefficient, a problem that can be remedied by splitting the neMD/MC moves into a two-step proposal/switch scheme [6]. The expensive switch component

Table 1: NAMD Performance and Efficiency

21M Atoms on 2,048 Nodes			
Machine	ns/day	ms/step	Efficiency
NCSA Blue Waters Cray XK7	42.1	4.1	65%
ALCF Theta Cray KNL	29.1	5.9	55%
NERSC Edison Cray XC30	28.9	6.0	81%
TACC Stampede KNC	20.4	8.5	46%
NCSA Blue Waters Cray XE6	19.5	8.8	85%
TACC Stampede CPU-only	16.5	10.5	79%
ALCF Mira Blue Gene/Q	4.9	35.1	81%

of this scheme can also be optimized using a well-defined efficiency metric based on the physical properties of the system [24]. All of these advances are now integrated into the constant-pH MD implementation in NAMD [23]. Overall, the new approach shows as good or better performance compared to previous efforts (Figure 2).

4 Code Development

The new implementation in NAMD developed as part of this ESP project is composed of three main parts:

1. Specific KNL optimizations for established NAMD functionality
2. New features for constant-pH MD
3. A new analysis application, `cphanalyze`, for constant-pH MD

The first and second items required the most effort and deserve the most comment. However, as a whole these components represent a specific new tool for the broader molecular simulation community.

4.1 Optimization for Theta

Optimizations were necessary at three levels of the machine hierarchy: the inter-node network, the within-node inter-core/inter-thread level, and the within-core vector level.

At the level of the inter-node network, the Charm++ runtime can utilize a fallback MPI layer, but generally achieves higher performance on Cray systems with a specialized GNI layer [29], initially developed for XE/KX machines with the Gemini torus network but equally effective on newer XC machines with the Aries dragonfly network. Network topology optimization was not attempted as it is generally not necessary on dragonfly networks. Due to the high core count of the KNL processor a “smp” build of Charm++ was selected, where multiple worker threads share a dedicated communication thread.

The recommended build options for Charm++ and NAMD on Theta are as follows:

```
./build charm++ gni-crayxc-persistent-smp --no-build-shared --with-production -xMIC-AVX512
```

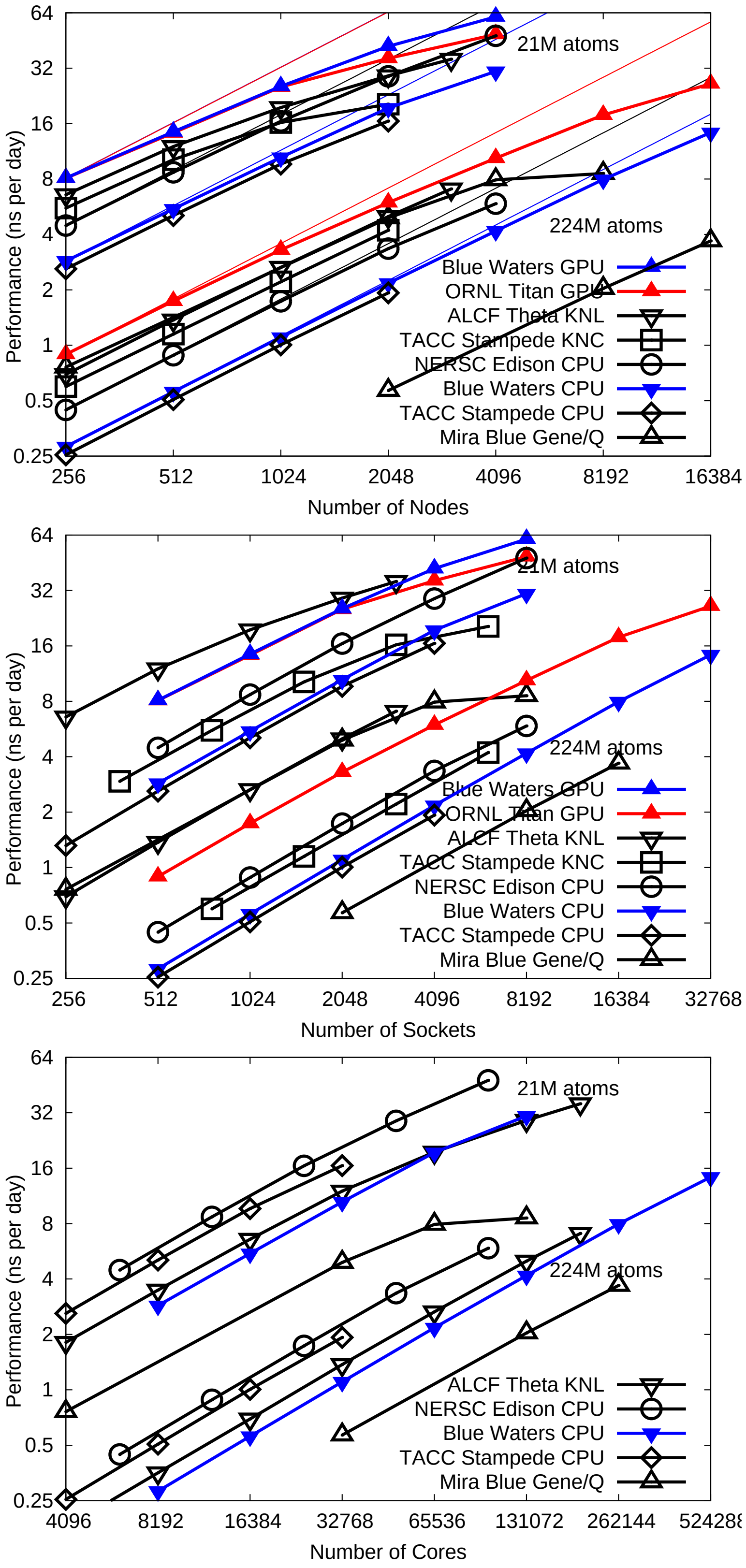

Figure 3: NAMD strong scaling on Theta and other machines plotted versus nodes, sockets, and cores for 21M and 224M atom benchmarks [22]. Theta per-node performance is comparable to NERSC Edison but does not scale as well. Theta per-socket (each CPU, GPU, or KNC considered one socket) performance is the best observed. Theta per-core (not including hardware threads) performance is comparable to Blue Waters. Compared to Mira, Theta is a factor of ten faster per node or per socket.

```
./config CRAY-XC-KNL-intel --with-fftw3 --charm-arch gni-crayxc-persistent-smp
```

Optimization for the core and thread structure of the KNL processor is achieved entirely through execution configuration options passed to either the Cray aprun command or Charm++ processor affinity options passed via the namd2 binary. The KNL processor is organized as pairs of cores that share a level-two cache, which each core supporting up to four hardware threads. The rules for selecting these configurations were as follows:

- one core (the last) is reserved for the operating system (via the aprun -r option);
- each communication thread should have a dedicated core;
- each level-two cache should be used only by threads of the same process.

Based on these rules and the number of cores of a given model of KNL processor (64 on Theta) only certain choices of processes-per-node and threads-per-process maximize both the number of worker threads for a given number of communication threads and the number of communication threads for a given number of worker threads, and any configuration that does not is assumed to be sub-optimal.

It was observed that either 1 or 2 hardware threads per core was optimal under different levels of available parallelism. It was also observed that multiple communication threads were needed per node. The resulting choices of optimal execution configurations for NAMD on Theta are shown below.

```
aprun -n $((4*$COBALT_JOBSIZE)) -N 4 -d 29 -j 2 -r 1 \
                                namd2 +ppn 14 +commap 14-62:16 +pemap 0-63:16.14
aprun -n $((4*$COBALT_JOBSIZE)) -N 4 -d 29 -j 2 -r 1 \
                                namd2 +ppn 28 +commap 14-62:16 +pemap 0-63:16.14+64
aprun -n $((7*$COBALT_JOBSIZE)) -N 7 -d 17 -j 2 -r 1 \
                                namd2 +ppn 8 +commap 56-62 +pemap 0-55
aprun -n $((7*$COBALT_JOBSIZE)) -N 7 -d 17 -j 2 -r 1 \
                                namd2 +ppn 16 +commap 56-62 +pemap 0-55+64
```

Charm++ pemap and commap options for worker and communication thread affinity are specified in a begin-end:stride.run+span format, where the "span" threads are interleaved in order to consecutively assign hyperthreads of the same core. As Theta uses 64-core KNL processors, two threads per core are used by adding "+64" to the end of the pemap configuration.

Optimization for the wide vector units of the KNL processor was achieved by the development of simplified, AVX-512 vectorizable version of the standard NAMD direct nonbonded kernel. The KNL-optimized kernel uses single-precision for force and energy calculation but double precision for force, energy, and virial accumulation. As in the NAMD CUDA kernels, single-precision atomic coordinates are stored relative to patch centers to avoid loss of precision as simulation sizes grow. While the standard NAMD kernel approximates all components of the pairwise interatomic potential via cubic interpolation as a function of r^2, the KNL kernel calculates the switched Lennard-Jones potential explicitly and interpolates the electrostatic force and energy both linearly as a function of r^{-1}. This interpolation method is copied from the NAMD CUDA kernel, and exploits the fast reciprocal operations available on both platforms.

The NAMD nonbonded kernel is broken into two stages for each outer-loop atom. The first stage filters an approximate pairlist, used for multiple timesteps, to an exact pairlist of only atoms within a fixed cutoff distance of the outer-loop atom. While the regular version of this function writes only the atom indices of the new list, the KNL version also writes the atomic coordinates and squared distances into separate aligned arrays (x, y, z, and r^2) so that they can be loaded as vectors in the force calculation stage. The loop is tagged `#pragma vector aligned` and `#pragma ivdep` to enable vectorization by the Intel compiler using gather instructions for input and compress instructions for output.

The second NAMD nonbonded kernel stage calculates and accumulates forces for the atoms selected by the first stage. This, much longer, stage is tagged `#pragma simd assert` along with multiple reduction tags for various energy and pressure components and the force on the outer-loop atom. Saved values from the previous stage are read via aligned loads, while gather and scatter instructions are generated for per-atom parameters such as charges and Lennard-Jones coefficients. As forces and energies are accumulated in double precision, compiler-generated assembly code was inspected to insure that only the minimum necessary number of precision conversions are performed. To avoid branches, a separate version of the kernel is built for each variant of the Lennard-Jones switching function.

The impact of the KNL-optimized kernel can be substantial, but fades as other performance-limiting factors become prominent. For example, for the standard 92K-atom apoa1 benchmark on 60 cores of a 64-core KNL, with one thread per core an 81% improvement is observed, but this drops to a 45% improvement with two threads per core, as with the optimized kernel the advantages of the second thread per core are outweighed by parallelization overhead.

The overall performance achieved is shown and compared to other machines in Fig. 3 and Table 1.

4.2 New Constant-pH MD features

As mentioned above, the constant-pH algorithm in NAMD is based on a hybrid neMD/MC scheme from early ideas by Stern [28], which were recently revised by Chen and Roux [6]. However, prior to 2015, there were no extant, documented implementations of either of these approaches that were appropriate for application on large-scale computing architectures. The implementation here is composed of both modifications to low-level C/C++ applications for performance sensitive activities as well as high-level routines using the NAMD-Tcl interface.

NAMD has long permitted on-the-fly control of MD simulations, whereby an input script can be used to pause a job instance, modify the simulation parameters in some way (possibly in response to some measured quantity), and then continue with the run. The new neMD/MC approach required two extensions of this ability. First, the simulation parameters must now include additional data structures to annotate not only the *current* protonation state of the system, but also other *possible* protonation states. Second, the simulation must be able to alternate between the standard equilibrium MD approach as well as new nonequilibrium approaches for driving the system between different protonation states. The new data structures for constant-pH MD are implemented in Tcl and direct changes to the molecular topology such that the protonation state is accurately represented. These routines leverage closer integration to the `psfgen` companion program as well as the new ability to reload the molecular topology on-th-fly. These advancements also permit repeatedly entering/exiting a nonequilibrium simulation mode via the Tcl interface.

4.3 New Analysis Code

Although the developments are not specific to Theta, we briefly describe a new utility, `cphanalyze`, which encapsulates our efforts to manage and analyze constant-pH MD specific output. In particular, constant-pH MD requires, at present, a new output format, `cphlog`, for tracking additional variables related to protonation states. In order to easily integrate data from multiple large-scale runs (a straightforward application of the previously developed NAMD multiple copy algorithm capabilities [15]), the new file format also includes a robust useage of meta-data. This design removes the burden of bookkeeping from the user and permits straightforward analysis of dozens or hundreds of files. For example, the following simple syntax seamlessly combines a "wild-card" selection of filenames from different simulations run in different directories:

```
$ cphanalyze */*.cphlog
```

There is nothing else the user needs to do. The utility also implements state-of-the-art multistate reweighting techniques [17, 27, 26, 30], which leads to considerably better results compared to conventional methods [23]. The code for this Python-based application is also made available with NAMD.

5 Portability

The port to Theta was relatively easy. The advantages in this were that all existing code did compile and execute as-is on the KNL processor with slow but not excessively poor performance, and that the Cray network and operating environment were carried over from other Cray XC machines to which NAMD had already been ported. The only aspects requiring attention were the greatly increased number of cores and threads per node and the longer AVX-512 vector instructions.

For portability going forward, although the optimized kernels are protected by `#pragma ifdef NAMD_KNL` macros they provide a performance benefit not only on the new Skylake processors with AVX-512 "core" instructions, but also for older AVX2 processors. Because the new kernels are designed for vectorization and rely only on a few pragmas, they should be easily adapted to other compilers and architectures, possibly requiring only translation to the standard OpenMPI 4.0 SIMD pragmas.

A downside of relying on the compiler to generate vector code from non-trivial loops is that performance and correctness are subject to bugs and regressions in future compiler versions. In fact this has already happened, as while the kernels were developed with the Intel 16.0 compilers available at the time, the Intel 17.0 compilers miss essential optimizations on Skylake and generate bad code for the original NAMD kernel on KNL (which is still used as a fallback for special cases). From a practical standpoint, it may be argued that in the face of unreliable compiler vectorization it may be less work in the long term to write kernels using explicit vector intrinsic functions, or some generic cross-platform version of such.

References

[1] Hans C. Andersen. Rattle: A "velocity" version of the Shake algorithm for molecular dynamics calculations. *J. Comput. Phys.*, 52:24–34, 1983.

[2] Axel Brünger, Charles L. Brooks III, and Martin Karplus. Stochastic boundary conditions in molecular dynamics simulations of ST2 water. *Chem. Phys. Lett.*, 105:495–500, 1984.

[3] Maike Bublitz, J. Preben Morth, and Poul Nissen. P-type ATPases at a glance. *J. Cell Sci.*, 124:2515–2519, 2011.

[4] Carlos A. Castañeda, Carolyn A. Fitch, Ananya Majumdar, Victor Khangulov, Jamie L. Schlessman, and Bertrand E. García-Moreno. Molecular determinants of the pK_a values of Asp and Glu residues in staphylococcal nuclease. *Proteins*, 77:570–588, 2009.

[5] Yunjie Chen and Benoît Roux. Efficient hybrid non-equilibrium molecular dynamics - Monte Carlo simulations with symmetric momentum reversal. *J. Chem. Phys.*, 141:114107, 2014.

[6] Yunjie Chen and Benoît Roux. Constant–pH hybrid nonequilibrium molecular dynamics–Monte Carlo simulation method. *J. Chem. Theory Comput.*, 11:3919–3931, 2015.

[7] Yunjie Chen and Benoît Roux. Generalized Metropolis acceptance criterion for hybrid non-equilibrium molecular dynamics–Monte Carlo simulations. *J. Chem. Phys.*, 142:024101, 2015.

[8] Gavin E. Crooks. Nonequilibrium measurements of free energy differences for microscopically reversible Markovian systems. *J. Stat. Phys.*, 90:1481–1487, 1998.

[9] Gavin E. Crooks. Entropy production fluctuation theorem and the nonequilibrium work relation for free energy differences. *Phys. Rev. E*, 60:2721–2726, 1999.

[10] Tom Darden, Darrin York, and Lee Pedersen. Particle mesh Ewald: An n log(n) method for Ewald sums in large systems. *J. Chem. Phys.*, 98:10089–10092, 1993.

[11] Ulrich Essmann, Lalith Perera, Max L. Berkowitz, Tom Darden, Lee Hsing, and Lee G. Pedersen. A smooth particle mesh Ewald method. *J. Chem. Phys.*, 103:8577–8593, 1995.

[12] David J. Hardy, Zhe Wu, James C. Phillips, John E. Stone, Robert D. Skeel, and Klaus Schulten. Multilevel summation method for electrostatic force evaluation. *J. Chem. Theory Comput.*, 11:766–779, 2015.

[13] Yangdong Huang, Wei Chen, Jason A. Wallace, and Jana Shen. All-atom continuous constant pH molecular dynamics with particle mesh Ewald and titratable water. *J. Chem. Theory Comput.*, 12:5411–5421, 2016.

[14] Christopher Jarzynski. Nonequilibrium equality for free energy differences. *Phys. Rev. Lett.*, 78:2690–2693, 1997.

[15] Wei Jiang, James C. Phillips, Lei Huang, Mikolai Fajer, Yilin Meng, James C. Gumbart, Yun Luo, Klaus Schulten, and Benoît Roux. Generalized scalable multiple copy algorithms for molecular dynamics simulations in NAMD. *Comput. Phys. Commun.*, 185:908–916, 2014.

[16] Werner Kühlbrandt. Biology, structure and mechanism of P-type ATPases. *Nat. Rev. Mol. Cell Biol.*, 5:282–295, 2004.

[17] Shankar Kumar, Djamal Bouzida, Robert H. Swendsen, Peter A. Kollman, and John M. Rosenberg. The weighted histogram analysis method for free-energy calculations on biomolecules. I. The method. *J. Comput. Chem.*, 13:1011–1021, 1992.

[18] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.

[19] Shuichi Miyamoto and Peter A. Kollman. SETTLE: An analytic version of the SHAKE and RATTLE algorithms for rigid water models. *J. Comput. Chem.*, 13:952–962, 1992.

[20] Jerome P. Nilmeier, Gavin E. Crooks, David D. L. Minh, and John D. Chodera. Nonequilibrium candidate Monte Carlo is an efficient tool for equilibrium simulation. *Proc. Natl. Acad. Sci. USA*, 108:E1009–E1018, 2011.

[21] James C. Phillips, Rosemary Braun, Wei Wang, James Gumbart, Emad Tajkhorshid, Elizabeth Villa, Christophe Chipot, Robert D. Skeel, Laxmikant Kalé, and Klaus Schulten. Scalable molecular dynamics with NAMD. *J. Comput. Chem.*, 26:1781–1802, 2005.

[22] James C. Phillips, Yanhua Sun, Nikhil Jain, Eric J. Bohm, and Laximant V. Kalé. Mapping to irregular torus topologies and other techniques for petascale biomolecular simulation. In *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, SC '14, pages 81–91. IEEE Press, 2014.

[23] Brian K. Radak, Christophe Chipot, Donghyuk Suh, Wei Jiang, James C. Phillips, Klaus Schulten, and Benoît Roux. Constant-pH molecular dynamics simulations for large biomolecular systems. *submitted.*

[24] Brian K. Radak and Benoît Roux. Efficiency in nonequilibrium molecular dynamics Monte Carlo simulations. *J. Chem. Phys.*, 145:134109, 2016.

[25] J. P. Ryckaert, G. Ciccotti, and H. J. C. Berendsen. Numerical integration of the Cartesian equations of motion of a system with constraints: Molecular dynamics of n-alkanes. *J. Comput. Phys.*, 23:327–341, 1977.

[26] Michael R. Shirts and John D. Chodera. Statistically optimal analysis of samples from multiple equilibrium states. *J. Chem. Phys.*, 129:124105, 2008.

[27] Marc Souaille and Benoît Roux. Extension to the weighted histogram analysis method: Combining umbrella sampling with free energy calculations. *Comput. Phys. Commun.*, 135:40–57, 2001.

[28] Harry A. Stern. Molecular simulation with variable protonation states at constant pH. *J. Chem. Phys.*, 126:164112, 2007.

[29] Yanhua Sun, Gengbin Zheng, L. V. Kale, Terry R. Jones, and Ryan Olson. A uGNI-based Asynchronous Message-driven Runtime System for Cray Supercomputers with Gemini Interconnect. In *Proceedings of 26th IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, Shanghai, China, May 2012.

[30] Zhiqiang Tan, Emilio Gallicchio, Mauro Lapelosa, and Ronald M. Levy. Theory of binless multi-state free energy estimation with applications to protein-ligand binding. *J. Chem. Phys.*, 136:144102, 2012.

[31] David E. Tanner, Kwok-Yan Chan, James C. Phillips, and Klaus Schulten. Parallel generalized Born implicit solvent calculations with NAMD. *J. Chem. Theory Comput.*, 7:3635–3642, 2011.

[32] M. E. Tuckerman, B. J. Berne, and G. J. Martyna. Reversible multiple time scale molecular dynamics. *J. Phys. Chem. B*, 97:1990–2001, 1992.

Argonne Leadership Computing Facility
Argonne National Laboratory
9700 South Cass Avenue, Bldg. #240
Argonne, IL 60439

www.anl.gov

Argonne National Laboratory is a U.S. Department of Energy
laboratory managed by UChicago Argonne, LLC